

SKRIPSI

**SISTEM PERAMALAN KENAIKAN PERMUKAAN AIR DENGAN
*ARTIFICIAL NEURAL NETWORKS BACKPROPAGATION***

**(Studi Kasus: Desa Tanjung Jaya, Kelurahan Surabaya, Kecamatan Sungai
Serut, Kota Bengkulu)**

Jenjang Strata I



Oleh

ILI SUSANTI

G1A010009

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS BENGKULU

2014

**SISTEM PERAMALAN KENAIKAN PERMUKAAN AIR DENGAN
*ARTIFICIAL NEURAL NETWORK BACKPROPAGATION***

**(Studi Kasus: Desa Tanjung Jaya, Kelurahan Surabaya, Kecamatan Sungai
Serut, Kota Bengkulu)**

SKRIPSI

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar
Sarjana Strata 1 (Satu) Pada Fakultas Teknik
Program Studi Teknik Informatika Universitas Bengkulu**



Oleh

ILI SUSANTI

G1A 010009

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS BENGKULU**

2014

MOTTO DAN PERSEMBAHAN

MOTTO

- ❖ *"Dimana ada kemauan di situ ada jalan, tidak ada kata tidak bisa selagi kita berusaha dan berdo'a"*
- ❖ *"Hidup adalah perjuangan"*

PERSEMBAHAN

Skripsi ini ku persembahkan untuk :

- ❖ *Nenek ku tersayang (Amidah) yang selalu mendo'akan ku agar cepat sukses*
- ❖ *Ayah dan Ibu tercinta (Saparudin dan Yati) yang selalu mendoakan dan mendorong semangat belajar Ku untuk mencapai keberhasilan.*
- ❖ *Saudara-saudaraku terkasih (Ayuk Septiani, Ayuk Gusti dan Adek Penti) yang selalu memberikan semangat dan motivasi.*
- ❖ *Liebe Mich (Sumarto Yudahirawan) yang selalu menjadi inspirasi dan motivator dalam hidup Ku.*
- ❖ *Sahabatku tersayang (Lisya Agusdina, Gita Triyana, Fatim Ulfah Karimah, Ancemona Yudha, Dyan Kemalasari) tempat curahan hatiku saat suka dan duka.*
- ❖ *Sahabat-sahabat terbaik Teknik dan Teknik Informatika 2010*
- ❖ *Amamaterku*

**SISTEM PERAMALAN KENAIKAN PERMUKAAN AIR DENGAN
ARTIFICIAL NEURAL NETWORK BACKPROPAGATION
(Studi Kasus : Desa Tanjung Jaya, Kelurahan Surabaya, Kecamatan Sungai
Serut, Kota Bengkulu)**

Oleh

Ili Susanti

NPM G1A01009

ABSTRAK

Penelitian ini bertujuan untuk meramalkan kenaikan permukaan air 1 (satu) jam kedepan. Dalam penelitian ini, sistem yang digunakan adalah sistem Jaringan Syaraf Tiruan *Backpropagation* dengan metode pelatihan dan pengujian *Levenberg-Marquardt*. Jaringan Syaraf Tiruan (JST) adalah model sistem komputasi yang berkembang sangat pesat yang dapat bekerja seperti syaraf biologis yaitu dapat mengenali pola-pola yang telah diajarkan. Langkah-langkah penelitian meliputi analisis kebutuhan sistem, pengumpulan data, perancangan perangkat lunak dan pengujian sistem. Pada penelitian ini, sebagai masukan JST adalah data koefisien aliran, intensitas curah hujan, dan luas daerah aliran sungai yang diolah menjadi kenaikan permukaan air selama satu hari (24 jam), sedangkan sebagai keluaran JST adalah kenaikan permukaan air untuk satu jam kemudian. Perangkat lunak ini dibangun dengan menggunakan bahasa pemrograman Matlab R2008b. Pengujian dilakukan untuk mencari arsitektur dan nilai parameter JST yang paling optimal pada proses pelatihan. Selain itu, pengujian dilakukan untuk mengetahui seberapa besar sistem mengenali data yang dimasukkan setelah melewati tahap pelatihan. Dari penelitian yang telah dilakukan, Jaringan Syaraf Tiruan dengan metode *Backpropagation* yang sudah dilatih dengan data koefisien aliran, intensitas curah hujan, dan luas daerah aliran sungai, dapat digunakan untuk memprediksi kenaikan permukaan air dengan persentase kesalahan 1.76625 %. Pada pengujian arsitektur dan parameter pelatihan, didapat bahwa sistem akan optimal dengan jumlah neuron 10 dan 5, laju pembelajaran 0.1 dan momentum 0.8.

Kata kunci: Kenaikan permukaan air, Jaringan Syaraf Tiruan, *Backpropagation*, *Levenberg-Marquardt*

***INCREASE IN SURFACE WATER SYSTEM PREDICTION WITH
ARTIFICIAL NEURAL NETWORKS BACKPROPAGATION
(Case Study : The village of Tanjung Jaya, Village of Surabaya, Serut River
District, the city of Bengkulu)***

By

Ili Susanti

NPM G1A010009

ABSTRACT

This study aims to predict the surface level rise of 1 (one) hour ahead. In this study, the system used a system with Backpropagation Neural Network training method and Levenberg-Marquardt testing. Artificial Neural Network (ANN) is a computational system model that is growing very rapidly which can work like biological neurons are able to recognize patterns that have been taught. Research steps include system requirements analysis, data collection, design and testing of software systems. In this study, the ANN is input the data flow coefficient, rainfall intensity, and the vast watershed that is processed into level rise for one day (24 hours), whereas the ANN output is the water level to rise one hour later. This software is developed using programming language Matlab R2008b. Testing is performed to find the architecture and parameter values on the optimal ANN training process. In addition, testing was conducted to determine how much the system recognize the data entered after passing the training phase. From the research that has been done, Neural Network with Backpropagation method that has been trained with the data flow coefficient, rainfall intensity, and the vast watershed, can be used to predict the flood discharge with an error percentage 1.76625%. In testing the architecture and training parameters, obtained that the system will be optimized by the number of 10 and 5 neurons, learning rate of 0.1 and momentum of 0.8.

Keyword: Increase in surface water, Artificial Neural Network, Backpropagation, Levenberg-Marquardt

KATA PENGANTAR



السَّلَامُ عَلَيْكُمْ وَرَحْمَةُ اللَّهِ وَبَرَكَاتُهُ

*Alhamdulillah*robbilalamin, penulis ucapkan kepada Allah SWT atas rahmat dan nikmat yang telah diberikan-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “ Sitem Peramalan Kenaikan Permukaan Air Dengan *Artificial Neural Network Backpropagation*”

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu (S1) di Program Studi Teknik Informatika Fakultas Teknik Universitas Bengkulu. Selesaiannya penulisan dan penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Pada kesempatan ini penulis mengucapkan terima kasih kepada :

1. Bapak Khairul Amri, S.T., M.T., sebagai Dekan Fakultas Teknik Universitas Bengkulu yang turut mendukung penulis untuk menyelesaikan skripsi ini.
2. Ibu Desi Andreswari, S.T., M.Cs sebagai Ketua Program Studi Teknik Informatika yang telah memberikan dukungan dan bantuan sehingga penulis dapat menjalankan setiap tahapan dengan baik dan lancar.
3. Bapak Drs. Asahar Johar, M. Si selaku dosen Pembimbing Akademik yang telah memberikan bimbingan kepada penulis dalam perkuliahan hingga penulis dapat menyelesaikan perkuliahan dengan baik.

4. Bapak Rusdi Efendi, S.T., M. Kom selaku dosen pembimbing utama yang selalu meluangkan waktu untuk memberikan bimbingan dan arahan yang sangat berharga kepada penulis dalam penyelesaian skripsi ini.
5. Bapak Dr. Gusta Gunawan, S.T., M.T selaku dosen pembimbing pendamping yang penuh kesungguhan, kecermatan dan kesabaran telah meluangkan waktu untuk memberikan bimbingan dan motivasi bagi penulis dalam penyelesaian skripsi ini.
6. Bapak Aan Erlansari, S.T., M.Eng dan Ibu Endina Putri Purwandari, S.T., M.Kom selaku dosen penguji pendamping yang telah memberikan dukungan dan bantuan sehingga penulis dapat menjalankan setiap tahapan dengan baik dan lancar, serta telah meluangkan waktu untuk menguji penulis.
7. Segenap Bapak dan Ibu Dosen Pengajar di Fakultas Teknik terutama Program Studi Teknik Informatika beserta staf administrasi di Fakultas Teknik Universitas Bengkulu.
8. Teman-teman seperjuangan Teknik Informatika angkatan 2010, kakak tingkat dan adik-adik tingkatku di Teknik dan Teknik Informatika Universitas Bengkulu yang telah banyak membantu dan memberikan motivasi dalam penyelesaian skripsi ini.
9. Semua pihak yang tidak bisa disebutkan satu persatu yang telah membantu penulis dalam menyelesaikan skripsi ini.

Penulis berharap semoga semua perhatian, motivasi, bimbingan dan dukungan yang telah diberikan menjadi amal dan mendapat balasan yang setimpal dari Allah SWT. Semoga skripsi ini dapat bermanfaat bagi semua pihak di masa yang akan datang.

وَلَسَّالَامُ عَلَيْكُمْ وَرَحْمَةُ اللَّهِ وَبَرَكَاتُهُ

Bengkulu, Juni 2014

Ili Susanti

G1A010009

DAFTAR ISI

HALAMAN JUDUL	i
MOTTO DAN PERSEMBAHAN	iv
ABSTRAK	vi
<i>ABSTRACT</i>	vii
KATA PENGANTAR.....	viii
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah	4
1.4 Tujuan dan Manfaat Penelitian	5
1.5 Sistematika Penulisan	5
BAB II LANDASAN TEORI	7
2.1 Curah Hujan.....	7
2.2 Bencana Banjir.....	7
2.3 Debit Banjir.....	8
2.4 Koefisien Aliran (C)	8
2.5 Jaringan Syaraf Tiruan (<i>Artificial Neural Network</i>)	10
2.5.1 Arsitektur Jaringan.....	13
2.5.2 Metode Pembelajaran.....	15
2.5.3 Fungsi Aktivasi	16
2.5.4 Algoritma <i>Backpropagation</i>	17
2.5.5 Algoritma <i>Levenberg-Marquardt</i>	23
2.6 Metode Pengembangan Sistem	25
2.7 Data Flow Diagram (DFD)	28
2.8 Diagram Alir (<i>Flowchart</i>).....	30
2.9 Perancangan <i>Database</i>	33
2.9.1 Relasi Antar Entitas (<i>Entity Relational</i>).....	33
2.10 MySQL	35
BAB III METODE PENELITIAN	36
3.1 Jenis Penelitian.....	36
3.2 Tempat dan Waktu Penelitian.....	37
3.3 Sarana Pendukung.....	37
3.4 Teknik Pengumpulan Data.....	37

3.5	Metode Pengembangan Sistem	39
3.6	Metode Pengujian	41
3.7	Jadwal Penelitian	41
BAB IV ANALISIS DAN PERANCANGAN SISTEM.....		43
4.1	Analisis Sistem.....	43
4.1.1.	Analisis Kebutuhan Sistem	44
4.1.2.	Analisis Data.....	45
4.1.3	Analisis <i>Flowchart</i>	46
4.2.	Perancangan Sistem	48
4.2.1.	Perancangan Data <i>Flow Diagram</i> (DFD).....	48
4.2.2.	Perancangan Antarmuka (<i>User Interface</i>)	52
4.3.	Struktur <i>Basis Data</i>	61
BAB V HASIL DAN PEMBAHASAN.....		63
5.1	Implementasi Sistem.....	63
5.2	Implementasi <i>Form</i> Menu Prediksi.....	64
5.3	Implementasi <i>Form</i> Bantuan.....	80
5.4.	Implementasi <i>Form</i> Menu Tentang.....	80
5.5	Implementasi <i>Form</i> Menu Keluar.....	81
5.6	Pengujian Sistem.....	82
5.7	Pengujian Parameter Jaringan.....	84
BAB VI KESIMPULAN DAN SARAN		89
6.1	Kesimpulan	89
6.2	Saran	90
DAFTAR PUSTAKA.....		91
LAMPIRAN.....		93

DAFTAR TABEL

Tabel 2.1 Nilai Koefisien Aliran Untuk Berbagai Penggunaan Lahan.....	9
Tabel 2.2 Perbandingan Hasil Pelatihan beberapa metode pelatihan JST dengan menggunakan data pelatihan yang sama.....	22
Tabel 2.3 Simbol-Symbol DFD	29
Tabel 2.4 Simbol-Symbol	32
Tabel 3.1. Jadwal Penelitian	42
Tabel 4.1. Data Latih	61
Tabel 4.2. Data Uji.....	61
Tabel 4.3. Data Ramal	61
Tabel 4.4. Data Validasi.....	62
Tabel 5.1. Perbandingan antara Data Target dengan Data Hasil Prediksi	78
Tabel 5.2. Perbandingan Data Target dan Data Hasil Prediksi tiap jam.....	82
Tabel 5.3. Pengujian Parameter Jaringan.....	84

DAFTAR GAMBAR

Gambar 2.1 Struktur Jaringan Syaraf Tiruan	13
Gambar 2.2 Model sekuensial linier	26
Gambar 4.1 Flowchart Proses Sistem Peramalan Kenaikan permukaan air secara umum	46
Gambar 4.2. Diagram Konteks	49
Gambar 4.3. DFD Level 1.....	50
Gambar 4.4 Rancangan Antarmuka <i>Form</i> Menu Utama	53
Gambar 4.5 Rancangan Antarmuka <i>Form</i> Prediksi	54
Gambar 4.6. Rancangan Antarmuka <i>Form</i> Bantuan.....	59
Gambar 4.7. Rancangan Antarmuka <i>Form</i> Tentang.....	60
Gambar 4.8. Rancangan Antarmuka <i>Form</i> Keluar	60
Gambar 5.1 Antarmuka <i>Form</i> Menu Utama.....	64
Gambar 5.2. Antarmuka <i>Form</i> Menu Prediksi	64
Gambar 5.3. Atribut-Atribut Pada <i>Form</i> Menu Prediksi	65
Gambar 5.4. Tampilan Proses <i>Training</i>	68
Gambar 5.5. Tampilan <i>Plot Performance</i>	69
Gambar 5.6. Tampilan <i>Plot Training State</i>	70
Gambar 5.7. Tampilan <i>Plot Regression</i>	71
Gambar 5.8. Hubungan antara target dengan <i>output</i> jaringan, untuk data pelatihan	72
Gambar 5.9. Perbandingan antara target dengan <i>output</i> jaringan, untuk data pelatihan..	73
Gambar 5.10. Hubungan antara target dengan <i>output</i> jaringan, untuk data pengujian	75
Gambar 5.11. Perbandingan antara target dengan <i>output</i> jaringan, untuk data pengujian	76
Gambar 5.12. Hasil Prediksi Kenaikan permukaan air pada Tanggal 7 Februari.....	77
Gambar 5.13. <i>Form Menu</i> Bantuan	80
Gambar 5.14. <i>Form Menu</i> Tentang.....	81
Gambar 5.15. <i>Form Menu</i> Keluar.....	81

DAFTAR LAMPIRAN

LAMPIRAN 1 Perhitungan Manual Data Pelatihan.....	A1-1
LAMPIRAN 2 Perhitungan Manual Data Uji.....	B1-1
LAMPIRAN 3 Perhitungan Manual Data Ramal.....	C1-1
LAMPIRAN 4 Pengujian Arsitektur Jaringan.....	D1-1

BAB I

PENDAHULUAN

1.1 Latar Belakang

Banjir adalah aliran/genangan air yang menimbulkan kerugian ekonomi atau bahkan menyebabkan kehilangan jiwa. Aliran/genangan air ini dapat terjadi karena adanya luapan-luapan pada daerah di kanan atau kiri sungai/saluran akibat alur sungai yang tidak memiliki kapasitas yang cukup bagi debit aliran yang lewat (Setyawan, 2008).

Cara penanggulangan banjir terbagi menjadi 2 (dua) yaitu upaya struktur dan non struktur. Upaya struktur antara lain pembangunan waduk, *floodway*, perbaikan alur sungai, retardasi (waduk alam). Dalam upaya pengendalian banjir, selain melalui upaya struktur dengan membangun bangunan prasarana pengairan pengendalian banjir, perlu juga dikembangkan upaya non struktur antara lain *Flood Planing Zooning* dan *Flood Forecasting Warning System* (FFWS) atau sistem prediksi dan peringatan dini banjir yang meliputi kegiatan prediksi (memperkirakan) besaran dan kapan akan terjadi banjir sekaligus pemberitahuan kepada masyarakat yang kemungkinan akan terjadinya banjir (Windarto, 2008).

Banjir dapat diprediksi dengan melakukan analisa kenaikan permukaan air di suatu tempat tertentu yang dianggap sebagai tempat terjadinya banjir. Proses perubahan naik turunnya kenaikan permukaan air di suatu daerah aliran sungai secara teoritis dapat dijelaskan, tetapi karena terdapat banyak variabel

maka model matematika sulit dibuat. Salah satu metode yang dapat digunakan adalah Jaringan Syaraf Tiruan atau *Artificial Neural Network* (ANN). Jaringan Syaraf Tiruan merupakan salah satu kecerdasan buatan yang merupakan representasi tiruan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Model ini merupakan model kotak hitam (*black box model*), sehingga dalam penerapannya tidak membutuhkan pengetahuan yang kompleks antar berbagai aspek dalam melakukan proses pemodelan faktor-faktor tersebut. Jaringan Syaraf Tiruan (*Artificial Neural Network*) terdapat beberapa metode pelatihan yakni diantaranya model *Hebb*, *Perceptron*, *Adaline*, *Backpropagation*, *Jaringan Kohonen*, *Radial Basis Function*, dan lain-lain.

Kelemahan Jaringan Syaraf Tiruan (JST) yang terdiri dari layar tunggal (seperti model *Hebb*, *Perceptron*, *Adaline*) membuat perkembangan JST menjadi terhenti pada tahun 1970 an. Penemuan *Backpropagation* yang terdiri dari beberapa layar membuka kembali cakrawala. Terlebih setelah berhasil ditemukannya berbagai aplikasi yang dapat diselesaikan dengan *Backpropagation*, membuat JST semakin diminati orang (Jek Siang, 2005).

JST dengan layar tunggal memiliki keterbatasan dalam pengenalan pola. Kelemahan ini bisa ditanggulangi dengan menambahkan satu/beberapa layar tersembunyi diantara layar masukan dan keluaran. Seperti halnya model JST lain, *Backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang

dipakai selama pelatihan. *Backpropagation* merupakan salah satu metode yang digunakan untuk peramalan dengan *time series* seperti penelitian yang dilakukan oleh Setyawan (2008) dalam skripsinya yang berjudul Peramalan Ketinggian Muka Air Dengan Jaringan Syaraf Tiruan Perambatan-Balik. Peramalan dengan *time series* berarti memprediksikan apa yang akan terjadi di masa datang berdasarkan pola deret data masa lalu. Oleh karena itu peramalan *time series* bertujuan memprediksikan apa yang akan terjadi, tanpa mengetahui mengapa hal itu terjadi. Sebab pada dasarnya peramalan ini memperlakukan sistem sebagai kotak hitam (*black box*) yang tidak diketahui mekanisme didalamnya.

Berdasarkan hasil pengamatan bahwa di Desa Tanjung Jaya, Kelurahan Surabaya, Kecamatan Sungai Serut terlihat hampir setiap tahun mengalami banjir. Kejadian ini berlangsung dari bulan Desember sampai dengan dengan bulan Maret. Pada periode bulan tersebut curah hujan yang terjadi cukup tinggi. Berdasarkan data intensitas curah hujan yang diperoleh dari BMKG pada tahun 2010, pada bulan Desember sampai dengan Maret, intensitas curah hujan rata-rata di daerah ini berkisar antara 3.7mm/jam sampai dengan 32.72mm/jam. Sementara daya tampung sungai di daerah ini berkisar antara 109.48 m³/ detik atau 0.03 m³/jam. Hal ini menyebabkan terjadi banjir pada tiap tahun.

Berdasarkan latar belakang diatas, maka penelitian mengenai “Peramalan Kenaikan Permukaan Air Dengan *Artificial Neural Networks Backpropagation*” penting untuk dilakukan.

1.2 Rumusan Masalah

Berdasarkan latar belakang dapat dirumuskan permasalahan adalah :
Bagaimana membangun suatu sistem informasi berbasis *artificial neural networks backpropagation* untuk meramalkan kenaikan permukaan air.

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini sebagai berikut:

1. Pada penelitian ini data yang akan digunakan sebagai data latih (*training*) adalah data koefisien aliran, luas daerah aliran sungai, dan data intensitas curah hujan yang digunakan adalah data sekunder dari BMKG dan dianggap sudah valid sehingga tidak dilakukan pengukuran ulang.
2. Model ini diujikan untuk meramalkan kenaikan permukaan air jangka pendek di suatu daerah aliran sungai (DAS) di wilayah hilir yaitu di Desa Tanjung Jaya, Kelurahan Surabaya, Kecamatan Sungai Serut, Kota Bengkulu.
3. Hasil dari penelitian ini adalah berupa grafik yang berisi mengenai informasi peramalan kenaikan permukaan air satu jam kemudian
4. Data masukan yang digunakan berupa data *time series* yang diimplementasikan kedalam *database* dengan menggunakan MySQL.
5. Untuk pembelajaran *Artificial Neural Network*, menggunakan *algoritma backpropagation* dengan metode pelatihan dan pengujian *Levenberg-Marquardt*.
6. Pada penelitian ini tidak menghitung pendangkalan kedalaman sungai

1.4 Tujuan dan Manfaat Penelitian

Tujuan dari tugas akhir ini adalah terbangunnya sistem perangkat lunak yang dapat digunakan untuk meramalkan besarnya kenaikan permukaan air dengan menggunakan *Artificial Neural Networks Backpropagation*.

Sedangkan manfaat dari penelitian ini mencakup 2 bagian yaitu sebagai berikut:

- 1) Bagi penulis, dapat mengimplementasikan ilmu yang diperoleh selama di bangku perkuliahan.
- 2) Bagi pengguna perangkat lunak, dapat memberikan kemudahan dalam meramalkan besar kenaikan permukaan air.

1.5 Sistematika Penulisan

Skripsi ini tersusun dari enam bab, tersusun sistematis dengan tujuan mempermudah pembacaan yang lebih akurat. Dalam penyusunan tugas akhir ini, sistematika penulisan dibagi menjadi beberapa bab sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang yang memunculkan suatu permasalahan sehingga dibutuhkan solusi untuk mengatasinya. Pada bab ini juga berisi rumusan masalah, batasan masalah, manfaat dan tujuan penelitian, serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi tinjauan pustaka yang berisikan teori-teori secara garis besar yang berhubungan dengan penelitian.

BAB III METODE PENELITIAN

Bab ini berisi tentang metode-metode yang digunakan dalam penelitian, seperti teknik pengumpulan data, metode pengembangan sistem, metode pengujian dan jadwal penelitian.

BAB IV ANALISIS DAN DESAIN PERANGKAT LUNAK

Bab ini menjelaskan setiap tahapan analisis dan perancangan sistem aplikasi yang akan dibangun dalam penelitian meliputi perencanaan sistem, analisis sistem, desain sistem dan implementasi sistem.

BAB V HASIL DAN PEMBAHASAN

Bab ini merupakan bab yang berisi hasil dan pembahasan yang menguraikan hasil perancangan sistem dan implementasinya.

BAB VI PENUTUP

Bab ini merupakan bab penutup yang merupakan bab terakhir yang berisi kesimpulan dan saran dari pembuatan tugas akhir sampai ke pengembangan perangkat lunak kedepannya.

BAB II

LANDASAN TEORI

Adapun teori yang digunakan sebagai dasar dalam penelitian ini antara lain teori tentang Bencana Banjir, Debit Banjir, Curah Hujan, Jaringan Syaraf Tiruan (*Artificial Neural Network*), Arsitektur Jaringan, Metode Pembelajaran, Fungsi Aktivasi, *Algoritma Backpropagation*, Metode Pengembangan Sistem, *Data Flow Diagram*, dan *Flowchart*.

2.1 Curah Hujan

Curah hujan adalah jumlah air yang jatuh di permukaan tanah datar selama periode tertentu yang diukur dengan satuan tinggi milimeter (mm) di atas permukaan horizontal. Dalam penjelasan lain curah hujan juga dapat diartikan sebagai ketinggian air hujan yang terkumpul dalam tempat yang datar, tidak menguap (*evaporasi*), tidak meresap (*infiltrasi*) dan tidak mengalir (Nasution, 2011).

2.2 Bencana Banjir

Banjir adalah aliran/ genangan air yang menimbulkan kerugian ekonomi atau bahkan menyebabkan kehilangan jiwa. Aliran/genangan ini dapat terjadi karena adanya luapan-luapan pada daerah di kanan atau kiri sungai/saluran akibat alur sungai tidak memiliki kapasitas yang cukup bagi debit aliran yang lewat (Setyawan, 2008).

Secara umum, terdapat tiga istilah pengertian banjir yang dikaitkan dengan sungai di masyarakat yaitu:

1. Suatu sungai dikatakan banjir apabila terjadi peningkatan debit aliran yang relatif besar, pengertian ini biasa digunakan oleh para petugas hidrologi dan masyarakat umum /awam setempat.
2. Suatu sungai dikatakan banjir apabila aliran air melimpas diluar alur sungai, pengertian ini biasa dipakai oleh instansi pengelola sungai /pengendali banjir.
3. Suatu sungai dikatakan banjir apabila aliran air melimpas ke luar alur sungai dan menimbulkan gangguan terhadap manusia. Pengertian ini biasa digunakan oleh media dalam kaitannya dengan informasi bencana banjir.

2.3 Debit Banjir

Debit banjir adalah aliran air yang berada di atas permukaan tanah sebagai akibat dari pengaruh curah hujan yang lebih besar dibandingkan dengan daya tampung sungai atau kapasitas sungai. Adapun rumus matematika dari debit banjir adalah sebagai berikut:

$$Q = C \cdot I \cdot A \dots \dots \dots (2.1)$$

Dimana:

Q = debit/aliran banjir (m³/ jam)

I = Intensitas hujan satuan (mm/jam)

A = Luas daerah aliran sungai (m²)

C = Koefisien aliran.

2.4 Koefisien Aliran (C)

Koefisien aliran permukaan (*runoff*) yang biasa dilambangkan dengan C didefinisikan sebagai nisbah antara laju puncak aliran permukaan terhadap

intensitas hujan. Faktor utama yang mempengaruhi nilai C adalah laju infiltrasi tanah, tanaman penutup tanah dan intensitas hujan. Adapun jika suatu Daerah Aliran Sungai (DAS) terdiri dari berbagai macam penggunaan lahan koefisien aliran permukaan yang berbeda, maka dapat dirumuskan sebagai berikut:

$$C_{das} = (C_1 A_1 + C_2 A_2 + C_3 A_3 + C_n A_n) / A_1 + A_2 + A_3 + A_n \dots \dots \dots (2.4)$$

Dimana:

$C_1 \dots C_n$ = koefisien aliran permukaan (ke-1 sampai dengan n)

A = luas lahan

Nilai Koefisien Aliran untuk Berbagai Penggunaan Lahan dapat dilihat pada Tabel 2.1

Tabel 2.1 Nilai Koefisien Aliran Untuk Berbagai Penggunaan Lahan

Tata guna lahan	Nilai c (%)	Penggunaan Lahan	Koefisien daerah aliran sungai
Kebun campuran	20	43,9564	8,79128
Perkebunan	40	2,749	1,0996
Taman, perkuburan	10	0,9228	0,09228
Pemukiman	50	299,8623	149,93115
Sawah	15	158,9403	23,841045
Semak	7	52,5601	3,679207
Tanah terbuka	25	11,807	2,95175
Tegalan	20	43,0518	8,61036
Hutan	50	92,8333	46,41665

(Sumber: Badan Pertanahan Nasional; Perhitungan Sendiri)

Berdasarkan Tabel 2.1 Nilai C_{das} untuk Desa Tanjung Jaya adalah sebagai berikut:

$$C_{das} = \frac{C_1 A_1 + C_2 A_2 + C_3 A_3 + C_4 A_4 + C_5 A_5 + C_6 A_6 + C_7 A_7 + C_8 A_8 + C_9 A_9}{A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7 + A_8 + A_9}$$

$$C_{das} = 0.2 (43.9564) + 0.4 (2.749) + 0.1 (0.9228) + 0.5 (299.8623) + 0.15 (158.9403) + 0.07 (52.5601) + 0.25 (11.807) + 0.2 (43.0518) + 0.5 (92.8333)$$

$$43.9564 + 2.749 + 0.9228 + 299.8623 + 158.9403 + 52.5601 + 11.807 + 43.0518 + 92.8333$$

$$C_{das} = 8.79128 + 1.0996 + 0.09228 + 149.93115 + 23.841045 + 3.679207 + 2.95175 + 8.61036 + 46.41665$$

$$706.683$$

$$C_{das} = \frac{245.413322}{706.683}$$

$$C_{das} = 0.347$$

$$C_{das} = 0.35$$

2.5 Jaringan Syaraf Tiruan (*Artificial Neural Network*)

Jaringan Syaraf Tiruan adalah sistem pemroses informasi dengan karakteristik dan performa yang mendekati syaraf biologis. Jaringan Syaraf Tiruan adalah generalisasi dari pemodelan syaraf biologi dengan asumsi-asumsi antara lain:

- a. Pemrosesan informasi terletak pada sejumlah komponen yang dinamakan neuron
- b. Sinyal merambat antara satu neuron ke neuron-neuron lainnya melalui jalur penghubung
- c. Tiap jalur penghubung memiliki bobot dan mengalikan besar nilai sinyal yang masuk (jenis neuron tertentu)

- d. Tiap neuron menerapkan fungsi aktivasi (biasanya nonlinear) yang menjumlahkan semua masukan untuk menentukan sinyal keluarannya

Selain memproses, jaringan syaraf tiruan juga memiliki kemampuan menyimpan informasi seperti definisi oleh (Widodo, 2013) bahwa jaringan syaraf adalah pemroses sederhana yang berjumlah banyak dan bekerja secara paralel dan terdistribusi. Jaringan ini memiliki kemampuan menyimpan pengetahuan dan memberikan saat dibutuhkan yang terdiri dari pengetahuan yang dimiliki sebagai hasil proses pembelajaran dan koneksi antar neuron yang berfungsi menyimpan pengetahuan itu. Oleh karena itu jaringan syaraf tiruan bermaksud membuat sistem yang menyerupai syaraf tiruan biologis.

Sedangkan menurut Trisnawan (2013) DARPA *Neural Network Study* (1988, AFCEA International Press) jaringan syaraf tiruan adalah sebuah jaringan syaraf yang dibentuk dari sejumlah elemen pemroses sederhana yang bekerja secara paralel dimana fungsinya ditentukan oleh stuktur jaringan, kekuatan hubungan, dan pengolahan dilakukan pada komputasi elemen atau *nodes*.

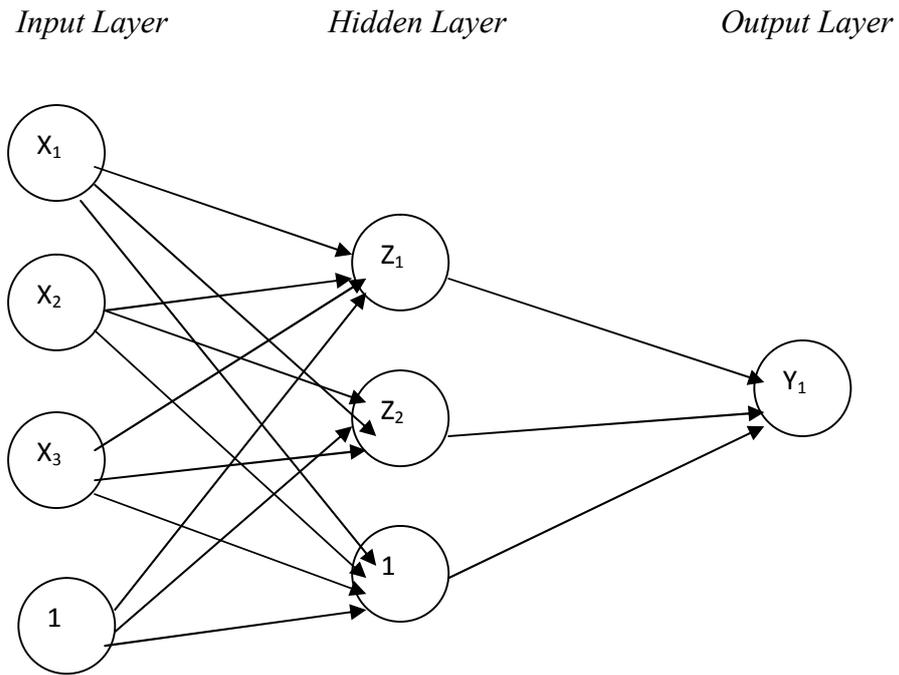
Jaringan syaraf tiruan memiliki kegunaan antara lain untuk perkiraan Fungsi, atau Analisis *Regresi*, termasuk prediksi *time series* dan modeling, klasifikasi, termasuk pengenalan pola dan pengenalan urutan, serta pengambil keputusan dalam pengurutan, pengolahan data, termasuk

penyaringan, pengelompokan, dan kompresi, dan program dari robot yang bergerak secara mandiri tanpa di gerakan oleh manusia.

Pemrosesan informasi pada jaringan syaraf tiruan ini terjadi pada elemen sederhana (*neuron*), kemudian sinyal dikirimkan diantara *neuron-neuron* melalui penghubung (*dendrit* dan *akson*) tetapi penghubung antar elemen memiliki bobot yang akan menambah atau mengurangi sinyal, selanjutnya untuk menentukan output, setiap neuron memiliki fungsi aktivasi (biasanya non linier) yang dikenakan pada semua *input* dan besar *output* akan dibandingkan dengan *threshold*.

Suatu model Jaringan Syaraf Tiruan (Irawan, 2012) ditentukan oleh:

- a. Pola antar neuron (arsitektur jaringan)
- b. Metode untuk menentukan dan mengubah bobot (disebut metode learning)
- c. Fungsi aktivasi
- d. Jaringan Syaraf Tiruan disebut juga: *brain metaphor, computational neuroscience, parallel distributed processing*.



Gambar 2.1 Struktur Jaringan Syaraf Tiruan
(Novikaginanto, 2012)

Berdasarkan Gambar 2.1 adalah *backpropagation* dengan 3 (tiga) *layer*, yakni *input layer*, *hidden layer*, dan *output layer*. *Input layer* terdiri dari x_1, x_2, x_3 yang merupakan data yang diinputkan, angka 1 (satu) merupakan bias pada *input layer*. Pada *hidden layer* terdapat z_1, z_2 yang merupakan penghubung antara sinyal *input* ke sinyal *output*. Angka 1 (satu) merupakan bias pada *hidden layer*. Pada *output layer* terdapat y_1 yang merupakan *output* dari *backpropagation* itu sendiri.

2.5.1 Arsitektur Jaringan

Jaringan Syaraf Tiruan memiliki beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi, antara lain: (Kusumadewi, 2004)

a. Jaringan layar tunggal

Jaringan dengan lapisan tunggal terdiri dari 1 (satu) *layer input* dan 1 (satu) *layer output*. Setiap neuron yang terdapat di *layer input* selalu terhubung dengan setiap neuron yang terdapat pada *layer output*. Jaringan ini hanya menerima *input* kemudian secara langsung mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi. Contoh algoritma yang menggunakan metode ini yaitu: ADALINE, *Hopfield*, *Perceptron*.

b. Jaringan layar jamak

Jaringan dengan layar jamak memiliki ciri khas tertentu yakni memiliki 3 (tiga) jenis *layer* yaitu *layer input*, *layer output* dan *layer* tersembunyi. Jaringan dengan banyak *layer* ini dapat menyelesaikan permasalahan yang lebih kompleks dibanding dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu yang cenderung lama. Contoh algoritma yang menggunakan metode ini yaitu: MADALINE, *Backpropagation*, *Necognitron*.

c. Jaringan layar kompetitif

Pada jaringan ini, sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif. Contoh algoritma yang menggunakan metode ini adalah *Learning Vector Quantization* (LVQ).

2.5.2 Metode Pembelajaran

Cara berlangsungnya pembelajaran atau pelatihan jaringan syaraf tiruan dikelompokkan menjadi 3 (tiga) yaitu: (Jek Siang, 2005)

a. *Supervised learning* (pembelajaran terawasi)

Pada metode ini, setiap pola yang diberikan kedalam jaringan syaraf tiruan telah diketahui *output*nya. Selisih antara pola *output* yang dihasilkan dengan pola *output* yang dikehendaki yang disebut *target error*. *Target error* digunakan untuk mengoreksi bobot jaringan syaraf tiruan sehingga jaringan syaraf tiruan mampu menghasilkan *output* yang sedekat mungkin dengan pola yang telah diketahui oleh jaringan syaraf tiruan. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini yakni : *Hebbian*, *Perceptron*, *ADALINE*, *Boltzman*, *Holpfield*, *Backpropagation*, *LVQ*.

b. *Unsupervised learning* (pembelajaran tak terawasi)

Pada metode ini, tidak memerlukan target *output*. Pada metode ini juga tidak dapat ditentukan hasil seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu tergantung pada nilai *input* yang diberikan. Tujuan pembelajaran ini adalah menglompokkan unit-unit yang hampir sama dalam area tertentu. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini yakni : *Kohonen*

c. *Hybrid learning* (pembelajaran hibrida)

Merupakan kombinasi dari metode pembelajaran *supervised learning* dan *unsupervised learning*. Sebagian dari bobotnya ditentukan melalui pembelajaran terawasi dan sebagian lainnya melalui pembelajaran tak terawasi. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini adalah RBF.

2.5.3 Fungsi Aktivasi

Dalam jaringan syaraf tiruan, fungsi aktivasi digunakan untuk menentukan keluaran suatu neuron. Argumen fungsi aktivasi adalah net masukan (kombinasi linier masukan dan bobotnya) (Jek Siang, 2005)

$$\text{Jika } net = \sum x_i w_i \dots\dots\dots(2.5)$$

maka fungsi aktivasinya adalah

$$f(net) = f(\sum x_i w_i) \dots\dots\dots(2.6)$$

Beberapa fungsi aktivasi yang sering digunakan adalah sebagai berikut :

a. Fungsi *threshold* (batas ambang)

$$f(x) = \begin{cases} 1, & \text{jika } x \geq \alpha \\ 0, & \text{jika } x < \alpha \end{cases} \dots\dots\dots(2.7)$$

Fungsi *threshold* merupakan fungsi *threshold* biner. Untuk kasus bilangan bipolar, maka angka 0 diganti dengan angka -1. Sehingga persamaan diubah menjadi

$$f(x) = \begin{cases} 1, & \text{jika } x \geq \alpha \\ -1, & \text{jika } x < \alpha \end{cases} \dots\dots\dots(2.8)$$

Adakalanya dalam jaringan syaraf tiruan ditambahkan suatu unit masukan yang nilainya selalu 1. Unit tersebut dikenal dengan bias. Bias dapat dipandang sebagai sebuah *input* yang nilainya selalu 1. Bias berfungsi untuk mengubah *threshold* menjadi 0.

b. Fungsi sigmoid

$$f(x) = \frac{1}{1+e^{-x}} \dots\dots\dots(2.9)$$

Fungsi ini sering digunakan karena nilai fungsinya yang sangat mudah untuk dideferensiasikan.

$$f'(x) = f(x)(1 - f(x)) \dots\dots\dots(2.10)$$

c. Fungsi identitas

$$f(x) = x \dots\dots\dots(2.11)$$

Digunakan jika keluaran yang dihasilkan oleh jaringan syaraf tiruan merupakan sembarang bilangan riil (bukan hanya pada range $[0,1]$ atau $[-1,1]$).

2.5.4 Algoritma *Backpropagation*

Algoritma *Backpropagation* meliputi 3 (tiga) fase. Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran. Fase

ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi.

a. Fase I : Propagasi maju

Selama propagasi maju, sinyal masukan (x_i) dipropagasikan ke layar tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit tersembunyi (z_j) tersebut selanjutnya dipropagasikan maju lagi ke layar tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan (y_k).

Berikutnya, keluaran jaringan (y_k) dibandingkan dengan target yang harus dicapai (t_k). Selisih antara $t_k - y_k$ adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.

b. Fase II : Propagasi mundur

Berdasarkan kesalahan $t_k - y_k$, dihitung faktor δ_k ($k = 1, 2, \dots, m$) yang dipakai untuk mendistribusikan kesalahan di unit y_k ke semua unit tersembunyi yang terhubung langsung dengan y_k . δ_k juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran.

Dengan cara yang sama, dihitung faktor δ_j di setiap unit di layar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga semua faktor δ di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

c. Fase III : Perubahan bobot

Setelah semua faktor δ dihitung, bobot semua garis dimodifikasi bersamaan. perubahan bobot suatu garis didasarkan atas faktor δ neuron di layar atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke layar keluaran didasarkan atas δ_k yang ada di unit keluaran.

Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan.

Berikut adalah penjelasan Algoritma *Backpropagation* (Fauset, 1993:294; (Widodo, 2013)):

Langkah 0. Inisialisasi bobot dari X_i ke Z_j dan dari Z_j ke Y_k dengan nilai acak serendah mungkin, set harga *error* minimal .

Langkah 1. Selama kondisi stop belum dilalui maka lakukan langkah 2-8.

Langkah 2. Untuk setiap pasang data masukan lakukan langkah 3-8.

Umpan maju (*feedforward*).

Langkah 3. Setiap unit sel masukan ($X_i, i=1, \dots, n$) menerima sinyal x_i dan menyebarkan sinyal ke seluruh lapis tersembunyi (*hidden layer*).

Langkah 4. Setiap unit sel tersembunyi ($Z_j, j=1, \dots, p$) menjumlahkan setiap sinyal masukan yang telah diberi bobot (v_{ij}) dan bias (v_{oj}),

$$z_in_j = v_{oj} + \sum_{i=1}^n x_i v_{ij}, \dots\dots\dots (2.12)$$

sinyal keluaran dihitung dengan fungsi aktivasi,

$$z_j = f(z_in_j), \dots\dots\dots (2.13)$$

dan mengirimkan hasilnya ke setiap unit sel di lapisan di atasnya (unit-unit keluaran).

Langkah 5. Setiap unit sel keluaran ($Y_k, k=1, \dots, m$) menjumlahkan setiap sinyal yang telah diberi bobot (w_{jk}) dan bias (w_{ok}),

$$y_in_k = w_{ok} + \sum_{j=1}^m z_j w_{jk}, \dots\dots\dots (2.14)$$

dengan fungsi aktivasi sinyal keluaran dihitung dengan,

$$y_k = f(y_in_k), \dots\dots\dots (2.15)$$

Perambatan balik nilai kesalahan (*backpropagation of error*).

Langkah 6. Setiap unit sel keluaran ($Y_k, k=1, \dots, m$) menerima pola target (t_k) yang sesuai dengan pola masukan pelatihan, hitung faktor δ ,

$$\delta_k = (t_k - y_k) f'(y_in_k), \dots\dots\dots (2.16)$$

hitung koreksi bobot dengan,

$$\Delta w_{jk} = \alpha \delta_k z_j, \dots\dots\dots (2.17)$$

hitung koreksi bias dengan,

$$\Delta w_{ok} = \alpha \delta_k, \dots\dots\dots (2.18)$$

selanjutnya nilai δ_k digunakan di lapisan dibawah.

Langkah 7. Setiap unit sel tersembunyi ($Z_j, j=1, \dots, k$) menjumlahkan nilai δ yang telah diberi bobot dari lapisan di atasnya,

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}, \dots\dots\dots(2.19)$$

Selanjutnya hitung nilai δ_j dengan,

$$\delta_j = \delta_{in_j} f'(z_{in_j}), \dots\dots\dots(2.20)$$

hitung koreksi bobot dengan,

$$\Delta w_{ij} = \alpha \delta_j x_i, \dots\dots\dots(2.21)$$

hitung koreksi bias dengan,

$$\Delta v_{0j} = \alpha \delta_j \dots\dots\dots(2.22)$$

Menghitung bobot baru.

Langkah 8. Setiap unit sel keluaran ($Y_k, k=1, \dots, m$) memperbaharui bobot bias dan lapisan dengan,

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}. \dots\dots\dots(2.23)$$

Setiap unit sel tersembunyi ($Z_j, j=1, \dots, p$) memperbaharui bobot bias dan lapisan dengan,

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \dots\dots\dots(2.24)$$

Langkah 9. Test kondisi stop. Kondisi stop yang digunakan adalah nilai Mean

Squre Error < toleransi, maksimum iterasi 1500, nilai toleransi 0,001.

Dimana:

x = masukkan (*input*)

j = 1 s/d n ($n = 10$)

v_{ij} = bobot pada lapisan tersembunyi

w_{ij} = bobot pada lapisan keluaran

n = jumlah unit pengolah pada lapisan tersembunyi

b = bias pada lapisan tersembunyi dan lapisan keluaran

k = jumlah unit pengolah pada lapisan keluaran

Y = *output*

δ_k = *error* pada lapisan keluaran

δ_j = *error* pada lapisan tersembunyi

v_{0j} = bias pada lapisan tersembunyi

w_{0j} = bias pada lapisan tersembunyi dan lapisan keluaran

Setelah proses pelatihan, *backpropagation* dapat digunakan untuk proses pengujian jaringan. Pada proses pengujian, tahap yang dilakukan hanya sampai tahap maju saja, tidak ada tahap mundur apalagi tahap modifikasi bobot. Seluruh bobot input diambil dari nilai bobot terakhir yang diperoleh dari proses pelatihan. Pada tahap pengujian ini, jaringan diharapkan dapat mengenali pola berdasarkan data baru yang diberikan (*generalisasi*).

Menurut Jaya (2007) terdapat beberapa metode pelatihan dan pengujian Jaringan Syaraf Tiruan dengan *backpropagation*. Hasil perbandingan beberapa metode yang terdapat pada pelatihan dan pengujian *backpropagation* dapat dilihat pada Tabel berikut:

Tabel 2.2 Perbandingan Hasil Pelatihan beberapa metode pelatihan JST dengan menggunakan data pelatihan yang sama. (Jaya, 2007)

No	Nama/penemu metode pelatihan JST Backpropagation (kusumadewi, 2004)	Nama Metode Pelatihan yang tersedia pada Toolbox Matlab	Jumlah Iterasi, Max 1500	MSE <0.001
1	Scaled Conjugate Gradient	Trainscg	53	$8,632 \times 10^{-04}$
2	Fletcher-Reeves Update	Traincgf	103	$9,421 \times 10^{-04}$
3	Polak-Ribie're	Traincgp	65	$9,224 \times 10^{-04}$
4	Powel-Beale Restars	Traincgb	47	$8,315 \times 10^{-04}$
5	Gradient Descent (GD) with Adaptive Learning Rate	Traingda	1167	$9,983 \times 10^{-04}$
6	GD with Momentum and Adaptive Learning Rate	Traingdx	117	$9,731 \times 10^{-04}$
7	Resilent Backpropagation	Trainrpb	210	$9,823 \times 10^{-04}$
8	Levenberg-Mawrqardt	Trainlm	11	$9,71 \times 10^{-05}$
9	Broyden, Fletcher, Goldfarb, Shanno	Trainbfg	28	$7,299 \times 10^{-04}$
10	One Step Secant	Trainoss	41	$9,155 \times 10^{-04}$

Berdasarkan Tabel diatas, terlihat bahwa metode pelatihan Levenberg-Marquardt memberikan nilai *Mean Square Error* (MSE) terkecil yaitu sebesar $9,71 \times 10^{-05}$ yang dihasilkan hanya dalam 11 kali hitungan iterasi. Ini berarti bahwa metode ini dapat mencapai fungsi tujuannya lebih cepat dibandingkan dengan metode lainnya.

2.5.5 Algoritma *Levenberg-Marquardt*

Algoritma *Levenberg-Marquardt* merupakan pengembangan dari algoritma *error backpropagation*. Algoritma ini dibangun untuk mengatasi beberapa kekurangan yang ada pada algoritma *error backpropagation* dengan memanfaatkan teknik optimasi numerik standar yaitu menggunakan pendekatan matriks *Jacobian*.

Tujuan dari *Levenberg-Marquardt* adalah meminimalkan total *error*. Berikut beberapa struktur algoritma *Levenberg-Marquardt*:

1. Inisialisasi bobot dan bias dengan bilangan acak, *epoch* maksimum, minimal *goal* (*performance* yang dihitung dengan MSE)
2. Menentukan parameter yang dibutuhkan, antara lain:
 - a. Parameter *Levenberg-Marquardt* yang nilainya harus lebih besar dari nol
 - b. Parameter faktor *input* dan bias yang digunakan sebagai parameter yang dikalikan atau dibagi dengan parameter *levenberg-Marquardt*
3. Menghitung maju (*feedforward*) pada *hidden* dan *otuput layer* seperti langkah-langkah pada algortima *error backpropagation*. Pada langkah (1)-(4).

4. Menghitung nilai *Mean Square Error* (MSE)

5. Menghitung *error* dan total *error* jaringan

a. Rumus untuk *error* :

$$e_r = t_r - y_r \dots\dots\dots(2.25)$$

r merupakan *input* ke- r

b. Rumus untuk menghitung total *error* :

$$e = [e_1 \ e_2 \ e_3 \ \dots \ e_N]^T \dots\dots\dots(2.26)$$

e merupakan vektor kesalahan berukuran $N \times 1$ yang terdiri dari e_r

$$r = 1, 2, 3, \dots, N$$

6. Menghitung matriks *Jacobian* $J(x)$

x merupakan matriks yang berisi nilai bobot dan bias dari keseluruhan jaringan.

$$X = [v_{11}, v_{12}, \dots, v_{ij}; v_{01}, v_{02}, \dots; w_{11}, w_{12}, \dots, w_{jk}; w_{01}, w_{02}, \dots, w_{0k}]$$

Matriks *Jacobian* berisi turunan pertama *error* jaringan terhadap bobot dan bias jaringan. Rumus untuk mencari *Jacobian* Matriks antara lain.

$$J = \left[\frac{\partial e_r}{\partial w} \right] \dots\dots\dots(2.27)$$

7. Setelah didapatkan nilai $J(x)$ maka dapat dihitung perubahan koreksi bobot dan biasnya dengan rumus berikut:

$$\Delta x = [J(x)^T J(x) + \mu I]^{-1} * \text{Gradient}$$

Gradient :

$$J(x)^T e \dots\dots\dots(2.28)$$

8. Setelah didapatkan nilai Δx tahap selanjutnya adalah pengkoreksian bobot dengan rumus yang sama seperti pada algoritma *error backpropagation*. Pada langkah 12
9. Menghitung maju (*feedforward*) dengan bobot dan bias yang baru. pada langkah (1-4)
10. Menghitung MSE jaringan dengan bobot dan bias yang baru. Kemudian tes kondisi berhenti
11. Jika epoch atau iterasi masih berlanjut maka akan terdapat 2 kemungkinan:
 - a. MSE naik
 - b. MSE turun
12. Kemudian lakukan kembali langkah 5 sampai dengan 8. (Amaliah, 2011)

Dimana:

e_r = *error* ke- r

t_r = *target* ke-r

y_r = *output* ke-r

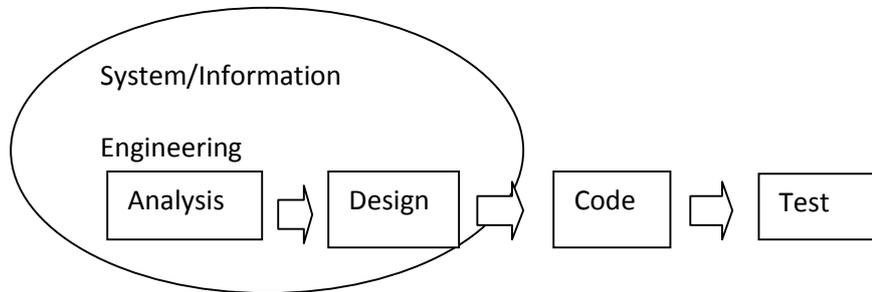
r = *input* ke-r

$J(x)$ = matriks Jacobian

2.6 Metode Pengembangan Sistem

Model sekuensial linier merupakan salah satu dari metode yang digunakan untuk pengembangan sistem. Sekuensial linier sering disebut juga dengan siklus kehidupan klasik atau model air terjun. Model sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan (Pressman,

2002). Gambar 2.2 menggambarkan model pengembangan sistem sikuensial linier.



Gambar 2.2 Model sekuensial linier (Pressman, 2002)

Model sekuensial linier melingkupi aktivitas-aktivitas sebagai berikut: (Pressman, 2002)

1. Rekayasa dan pemodelan sistem

Karena sistem merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan ke *software* tersebut. Pandangan sistem ini penting ketika *software* harus berhubungan dengan elemen-elemen yang lain seperti *software*, manusia, dan database. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta disain tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.

2. Analisis kebutuhan *software*

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada *software*. Untuk memahami sifat program yang

dibangun, analisis harus memahami domain informasi, tingkah laku, unjuk kerja, dan *interface* yang diperlukan. Kebutuhan baik untuk sistem maupun *software* didokumentasikan dan dilihat lagi dengan pelanggan.

3. Desain

Desain *software* sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda, struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural. Proses desain menterjemahkan syarat/kebutuhan ke dalam sebuah representasi *software* yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi *software*.

4. Generasi kode

Desain harus diterjemahkan kedalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis. Implementasi program dalam Tugas Akhir ini penulis mengimplementasikan desain ke dalam bentuk bahasa pemrograman. Bahasa pemrograman yang digunakan adalah bahasa pemrograman *Matlab* R2008b.

5. Pengujian

Sekali program dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal *software*, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional, yaitu

mengarahkan pengujian untuk menemukan kesalahan–kesalahan dan memastikan bahwa *input* yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

6. Pemeliharaan

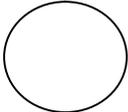
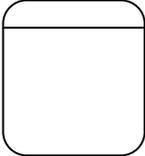
Software akan mengalami perubahan setelah disampaikan kepada pelanggan (perkecualian yang mungkin adalah *software* yang dilekatkan). Perubahan akan terjadi karena kesalahan–kesalahan ditentukan, karena *software* harus disesuaikan untuk mengakomodasi perubahan–perubahan di dalam lingkungan eksternalnya (contohnya perubahan yang dibutuhkan sebagai akibat dari perangkat peripheral atau sistem operasi yang baru), atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan *software* mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.

2.7 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama *Bubble chart*, *Bubble diagram*, model proses, diagram alur kerja, atau model fungsi (Wijyaning, 2009).

Symbol yang digunakan pada DFD diperoleh dari dua sumber, yaitu DeMarco & Yourdan Symbols dan Gane & Sarson Symbols. Symbol dari kedua sumber tersebut sama-sama dapat digunakan, hanya saja harus konsisten dalam penggunaannya. Adapun simbol-simbol yang disebutkan tersebut ditunjukkan pada Tabel 2.3 sebagai berikut:

Tabel 2.3 Simbol-Simbol DFD (Wijyaning, 2009)

DeMarco & Yourdan Simbol	Gane & Sarson Simbol	Keterangan
		Entitas Luar (Terminator)
		Proses
		Data <i>flow</i> (arus data)
		Data <i>store</i> (simpan data)

Dalam DFD terdapat proses perubahan *input* menjadi *output*. Untuk mempermudah pembacaan DFD dibuat bertingkat/level dari mulai level 0 sampai dengan level yang diperlukan antara lain (Wijyaning, 2009).

1. Diagram Konteks (Diagram Level Nol)

Diagram konteks merupakan level tertinggi dari DFD, yang memperlihatkan sistem sebagai sebuah proses dan menggambarkan hubungan sistem dengan lingkungan luarnya. Tujuannya adalah memberikan pandangan umum mengenai sistem.

2. Diagram Level Satu

Diagram level satu adalah turunan langsung dari diagram konteks yang menjelaskan proses secara lebih terperinci.

3. Diagram level Dua, Tiga, dan seterusnya.

Diagram ini merupakan dekomposisi dari level sebelumnya. Proses dekomposisi dilakukan sampai dengan proses siap dituangkan ke dalam program.

2.8 Diagram Alir (*Flowchart*)

Menurut Jogiyanto (2005) “Diagram alir (*flowchart*) adalah suatu bagan (*chart*) yang menunjukkan aliran (*flow*) dalam program atau prosedur sistem secara logika”. Jadi, pembuatan *flowchart* ini digunakan untuk menggambarkan sistem baru yang akan dikembangkan secara logis tanpa mempertimbangkan terlebih dahulu lingkungan fisik dimana sistem ini akan digunakan.

Ada beberapa hal yang perlu diperhatikan pada waktu menggambar atau membuat *flowchart*, yaitu :

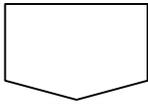
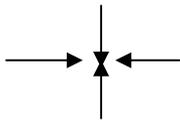
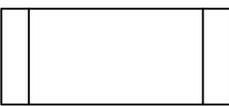
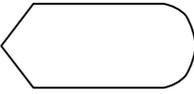
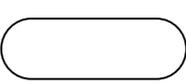
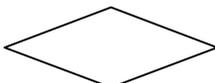
1. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan dalam bagan alir harus ditunjukkan dengan jelas.
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhir.

4. Masing-masing kegiatan di dalam bagan alir harus di dalam urutan yang semestinya.
5. Kegiatan yang terpotong akan disambung di tempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
6. Gunakan simbol-simbol bagan alir yang standar.

Symbol-simbol yang digunakan pada *flowchart* akan ditunjukkan pada Tabel

2.4

Tabel 2.4 Simbol-Simbol *Flowchart* (Syarif, 2009)

No	Simbol	Fungsi
1		<i>Off page connector</i> , merupakan simbol masuk atau keluarnya suatu prosedur pada lembar kertas lain
2		Arus atau <i>flow</i> , prosedur yang dapat dilakukan dari atas ke bawah, bawah ke atas, dari ke kanan atau dari kanan-kanan ke kiri
3		<i>Document</i> merupakan simbol untuk data yang berbentuk informasi
4		<i>Predefined process</i> , untuk menyatakan sekumpulan langkah proses yang ditulis sebagai prosedur
5		Simbol untuk <i>output</i> yang ditujukan suatu <i>divice</i> , seperti <i>printer</i> , <i>plotter</i>
6		Untuk menyimpan data
7		<i>Predefined process</i> , suatu simbol untuk menyediakan tempat-tempat untuk pengolahan data dalam <i>storage</i>
8		<i>Connector</i> , suatu prosedur akan masuk atau keluar melalui simbol ini dalam lembar yang sama
9		Terminal, untuk memulai dan mengakhiri suatu program
10		Proses, suatu simbol yang menunjukkan setiap pengolahan yang dilakukan oleh komputer
11		<i>Input-output</i> , untuk memasukkan data maupun menunjukkan hasil dari suatu proses.
12		<i>Decision</i> , Suatu kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau pilihan

2.9 Perancangan *Database*

2.9.1 Relasi Antar Entitas (*Entity Relational*)

Model *Entity-Relationship* digunakan untuk menyajikan objek data secara visual dalam bentuk diagram (Pratiwi, 2011). Berikut ini merupakan dasar-dasar membangun pemodelan E-R.

1. Entitas

Entitas adalah objek data prinsip tentang informasi yang dikumpulkan. Entitas digolongkan menjadi *independent* atau *dependent* (dalam beberapa metodologi, istilah yang digunakan adalah lemah dan kuat secara berturut-turut). Suatu *independent entity* adalah apa yang tidak bersandar pada yang lain sebagai identifikasi.

2. Atribut

Atribut menguraikan entitas di mana mereka dihubungkan. Kejadian dari suatu atribut tertentu adalah suatu nilai. Atribut bisa digolongkan sebagai *identifiers* atau *descriptors*. *Identifiers* lebih umum disebut kunci, yang secara unik mengidentifikasi suatu kejadian dari suatu entitas. *Descriptor* menguraikan suatu karakteristik yang tidak unik dari suatu kejadian entitas.

3. Relasi

Hubungan disesuaikan dengan derajat, konektivitas, kardinalitas, arah, jenis, dan keberadaan. Tidak semua metodologi pemodelan menggunakan kriteria tersebut.

Derajat relasi adalah sejumlah entitas yang berhubungan dengan hubungan. Hubungan n-ary adalah format yang umum untuk derajat n.

Relasi dalam basisdata relasional mempunyai beberapa karakteristik, yaitu:

- a. Semua elemen data pada suatu baris dan kolom tertentu harus mempunyai nilai tunggal (*single value*), atau suatu nilai yang tidak dapat dibagi-bagi (*atomic value*), bukan suatu larik atau grup perulangan.
- b. Semua *entry*/elemen data pada suatu kolom tertentu dalam relasi yang sama harus mempunyai jenis yang sama. Masing-masing kolom dalam suatu relasi mempunyai suatu nama yang unik (meskipun kolom-kolom dalam relasi yang berbeda diizinkan mempunyai nama yang baik).

4. Kardinalitas

Kardinalitas relasi dapat ditunjukkan dengan banyaknya garis cabang atau dengan pemakaian angka, yaitu: (Pratiwi, 2011)

- a. Relasi satu ke satu (1 dan 1)
- b. Relasi satu ke banyak (1 dan m)
- c. Relasi banyak ke satu (m dan 1)
- d. Relasi banyak ke banyak (m dan m)

2.10 MySQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. MySQL termasuk jenis RDBMS (*Relationa Database Management System*). Pada MySQL, sebuah *database* mengandung satu atau sejumlah Tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. Untuk mengelola *database* MySQL ada beberapa cara yaitu melalui prompt DOS (*tool command line*) dan dapat juga menggunakan program utility seperti: (Pratiwi, 2011)

1. PHP MyAdmin
2. MySQL GUI
3. MySQL Manager Java Based
4. MySQL Administrator for windows

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian yang digunakan dalam penelitian ini adalah penelitian terapan. Penelitian terapan adalah penyelidikan yang hati-hati, sistematis dan terus menerus terhadap suatu masalah dengan tujuan untuk digunakan dengan segera untuk keperluan tertentu (Soemarno, 2007). Hasil dari penelitian terapan tidak perlu satu penemuan baru, tapi merupakan aplikasi baru dari penelitian yang telah ada. Penelitian ini berusaha menerapkan teori atau metode yang telah dikembangkan baik dalam cakupan penelitian murni maupun penelitian terapan seperti sistem basis data, bahasa pemrograman, dan lain-lain.

Pada penelitian ini akan dibangun suatu sistem yang dapat meramalkan besar kenaikan permukaan air dalam jangka pendek (satu jam kedepan) di Desa Tanjung Jaya, Kelurahan Surabaya, Kecamatan Sungai Serut, Kota Bengkulu. Dengan menggunakan metode Jaringan Syaraf Tiruan (*Artificial Neural Network*) dengan model Perambatan-Balik (*Backpropagation*) dengan metode *Levenberg-Marquardt* sebagai metode pelatihan dan pengujian.

3.2 Tempat dan Waktu Penelitian

Penelitian ini dilakukan dengan cara dokumentasi dan kepustakaan. Sumber data dari penelitian ini berupa hasil dokumentasi, seperti data intensitas curah hujan diperoleh dari Badan Meteorologi, Klimatologi dan Geofisika (BMKG) Provinsi Bengkulu selama 4 bulan (bulan November 2013 – Februari 2014). Data luas Daerah Aliran Sungai (DAS) diambil dari BPDAS Ketahun, dan data tata guna lahan yang diambil dari Badan Pertanahan Nasional (BPN).

3.3 Sarana Pendukung

Sarana pendukung yang digunakan untuk melakukan penelitian ini, adalah sebagai berikut:

1. Perangkat Keras (*Hardware*) yang digunakan adalah:
 - a. Laptop Acer dengan spesifikasi Processor Intel Pentium P6200, RAM 1 GB, Hard Disk 500 GB.
 - b. Printer Cannon iP 2770
2. Perangkat Lunak (*Software*) yang digunakan adalah:
 - a. Sistem Operasi Windows Seven Ultimate
 - b. Matrix Laboratory (MatLab) R2008b

3.4 Teknik Pengumpulan Data

Teknik pengumpulan data yang dilakukan pada penelitian ini adalah dengan melakukan pengumpulan data hasil dokumentasi dan

kepastakaan. Dokumentasi yaitu mengumpulkan data-data berdasarkan hasil penelitian atau perhitungan yang telah dilakukan sebelumnya.

Studi kepastakaan yaitu pengumpulan data yang diperoleh dari buku-buku, karya ilmiah, pendapat para ahli yang memiliki relevansi dengan Sistem Peramalan Kenaikan permukaan air Dengan *Artificial Neural Network Backpropagation* (Studi Kasus : Desa Tanjung Jaya, Kelurahan Surabaya, Kecamatan Sungai Serut, Kota Bengkulu). Studi kepastakaan yang digunakan dalam penelitian ini berupa:

a. Buku

Buku yang digunakan dalam penelitian ini berupa buku-buku referensi yang dapat menunjang materi mengenai jaringan syaraf tiruan (*artificial neural network*), pemrograman *Graphic User Interface* (GUI) dengan MatLab, sintaks-sintaks MatLab, dan buku elektronik yang didapat di internet yang berhubungan dengan penelitian ini.

b. Artikel

Artikel yang digunakan sebagai sumber data adalah artikel yang didapat dari internet yang berhubungan dengan penelitian ini.

c. Jurnal dan Skripsi

Bahan ilmiah lain yang digunakan sebagai acuan dalam penelitian ini adalah Jurnal dan Skripsi di bidang yang relevan dengan penelitian ini, yaitu mengenai sistem peramalan atau pembahasan mengenai metode *Backpropagation*.

3.5 Metode Pengembangan Sistem

Sistem yang dikembangkan dalam penelitian ini menggunakan model pengembangan sistem *sekuensial linier* yang bersifat sistematis dan berurutan. Adapun penjelasan tahap-tahap model *sekuensial linier* dalam penelitian ini adalah sebagai berikut:

1. Rekayasa dan Pemodelan sistem

Karena sistem merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan *software* tersebut. Pandangan sistem ini penting ketika *software* harus berhubungan dengan elemen-elemen yang lain seperti *software*, manusia, dan *database*. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta disain tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.

2. Analisis kebutuhan sistem

Pada tahap ini peneliti akan melakukan analisis dan definisi kebutuhan sistem dengan teknik pengumpulan data menggunakan dokumentasi, serta mengumpulkan semua data yang bersumber dari literatur berupa buku-buku, laporan penelitian, karangan-karangan ilmiah dan lain sebagainya mengenai hal-hal yang dibutuhkan dan mendukung proses pembuatan sistem peramalan kenaikan permukaan air ini. Setelah itu, dilakukan analisis sistem yang akan dibangun. Hasil analisis ini akan digambarkan secara terstruktur yaitu dengan membuat DFD. Langkah

selanjutnya adalah membuat *flowchart* untuk menunjukkan aliran (*flow*) dalam program atau prosedur sistem secara logika.

3. Desain Sistem

Desain *software* sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda, yaitu struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural. Proses desain menterjemahkan syarat/kebutuhan ke dalam sebuah representasi *software* yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi *software*.

4. Generasi Kode

Desain harus diterjemahkan kedalam bentuk yang dimengerti oleh mesin yaitu ke dalam bahasa pemrograman yang telah ditentukan melalui proses penulisan program (*coding*). Dalam penelitian ini, digunakan MatLab[®] R2008b

5. Pengujian Sistem

Sistem yang sudah dibangun akan dilakukan pengujian untuk melihat apakah sistem tersebut sesuai dengan perencanaan dan perancangan.

6. Pemeliharaan

Tahap ini adalah tahap akhir pengembangan dan implementasi sistem yaitu pengoperasian sistem secara nyata. Namun dalam pengoperasiannya tetap dibutuhkan dukungan agar sistem dapat digunakan dalam jangka panjang dengan melakukan pemeliharaan sistem.

Pemeliharaan sistem dilakukan bukan hanya sekedar proses memperbaiki kesalahan program tetapi proses yang memiliki karakteristik memperbaiki kesalahan yang tidak ditemukan pada tahapan sebelumnya atau menambahkan fungsi baru yang belum ada pada program tersebut.

3.6 Metode Pengujian

Pada penelitian ini, akan dilakukan 2 (dua) jenis pengujian. Pengujian yang pertama dilakukan untuk menguji tingkat akurasi dari peramalan secara keseluruhan. Kedua pengujian dilakukan untuk melihat berapa selisih *error* antara hasil peramalan dengan data sesungguhnya dengan menggunakan MAPE (*Mean Absolute Percentange Error*). Berikut adalah rumus untuk menghitung MAPE dan rumus tingkat akuasi dapat ditulis sebagai berikut:

$$\text{Tingkat akurasi(\%)} = \frac{\Sigma \text{data uji benar}}{\Sigma \text{jumlah total data uji}} \times 100\% \dots\dots\dots(3.1)$$

$$\text{MAPE(\%)} = \frac{1}{N} \sum_{i=1}^N (P_A^i - P_F^i / P_A^i) \times 100\% \dots\dots\dots(3.2)$$

Dimana:

P_A = data aktual

P_F = data hasil peramalan

N = Jumlah data

3.7 Jadwal Penelitian

Berikut adalah jadwal penelitian yang akan dilaksanakan dapat dilihat pada Tabel 3.1 di bawah ini:

Tabel 3.1. Jadwal Penelitian

No	Kegiatan	Bulan/Tahun					
		Feb	Mar	Apr	Mei	Juni	Juli
1.	Pengajuan Judul Proposal	■					
2.	Penyusunan Proposal	■					
3.	Seminar Proposal		■				
4.	Perbaiki Proposal		■				
5.	Pengumpulan Data		■				
6.	Analisis dan Perancangan Sistem		■				
7.	Pembuatan Koding		■	■			
8.	Implementasi dan Pengujian Unit			■	■		
9.	Integrasi dan Pengujian				■		
10.	Analisis Hasil				■		
11.	Sidang Hasil					■	
12.	Revisi Skripsi					■	
13.	Final Buku Skripsi						■